

Prova Prática de Estrutura de Dados

- Baixar materiais: 20 minutos
- Realizar a prova: 80 minutos

Instruções:

- A prova contém 3 exercícios distintos.
- É permitido consultar anotações e materiais do curso, porém não pode usar a internet.
- Os exercícios devem ser implementados em C++.

(até 3 pontos) Exercício 1 – Verificador de Palíndromos com Lista Duplamente Encadeada

Implemente um programa que use uma lista duplamente encadeada para verificar se uma palavra é um palíndromo (lê-se igual de trás pra frente).

- O programa deve solicitar uma palavra de tamanho ímpar.
- Armazene os caracteres em uma lista duplamente encadeada.
- Compare os elementos do início e do fim simultaneamente, até o centro da lista.
- Informe se é ou não um palíndromo.

Exemplos:

Entrada:	Saída:
arara	É um palíndromo!
professor	Não é um palíndromo!

Implemente as seguintes funções:

- `inserir_inicio(elemento)`
- `inserir_fim(elemento)`
- `remover_inicio()`
- `remover_fim()`

(até 4 pontos) Exercício 2 – Estrutura de Árvore

Implemente uma Árvore Binária de Busca (BST) para armazenar os meses do ano, ordenados alfabeticamente. Você deve implementar as seguintes funcionalidades:

- Uma função `inserir_mes(nome_mes)` que insere o nome do mês na árvore.

- Uma função `exibir_em_ordem()` que imprime os meses em ordem alfabética.
- Uma função `exibir_pre_ordem()` que imprime os meses em pré-ordem.

Personalização obrigatória:

Nos métodos de exibição, o mês correspondente ao seu aniversário (com base no mês de nascimento) deve ser destacado com a mensagem "(parabéns para mim)" ao lado.

Exemplo:

abril, agosto, novembro (parabéns para mim), outubro

(até 3 pontos) Exercício 3 – Identificação de Rotações em uma Árvore AVL

Crie um programa que receba três números inteiros, simulando sua inserção sequencial em uma árvore AVL que já possui números inseridos nessa ordem:

- 2 primeiros números do seu prontuário.
- 2 próximos números do seu prontuário.
- 2 próximos números do seu prontuário.
- Último número do seu prontuário.
- Dia de nascimento (só inserir se não estiver sido inserido anteriormente)
- Mês de nascimento (só inserir se não estiver sido inserido anteriormente)
- 2 primeiros números ano de nascimento (só inserir se não estiver sido inserido anteriormente)
- 2 últimos números ano de nascimento (só inserir se não estiver sido inserido anteriormente)

Comentar no código quais números foram inseridos na árvore.

Exemplo:

// 26, 58, 98, 7, 25, 11, 19, 98 (já foi inserido anteriormente)

Com base na ordem de inserção, o programa deve identificar qual tipo de rotação seria necessária para balancear a árvore, dentre as opções:

- LL (rotação simples à direita)
- RR (rotação simples à esquerda)
- LR (rotação dupla: esquerda-direita)
- RL (rotação dupla: direita-esquerda)
- Nenhuma (caso já esteja balanceada)

Dicas:

- Os problemas são mais simples do que parecem.
- Tem diversas formas diferentes de resolver cada problema, se você não lembra algum código, busque alternativas para programar.
- Se não lembrar como se digita alguma função em C++ pode perguntar ao professor, se fizer sentido ele vai falar como fazer.

#include <string>	pacote para trabalhar com string
.length()	retorna o tamanho da string
for (char c : palavra)	percorrer os caracteres de uma palavra